**Rabbitmq 3.11.11 K8S上集群部署文档操作手册**

**一、先加载镜像到用户环境**

1、master节点 rabbtimq解压目录下操作（产线rabbitmq目录，项目rabbitmq\_3.11.11-demo目录）：

docker load -i rabbitmq\_cluster3.11.11-v1.tar

2、备注：需要修改为用户镜像，再push前执行 docker tag registry.edoc2.com:5000/edoc2v5/rabbitmq\_cluster:3.11.11-v1 {用户指定的image tag}

默认执行：

docker push registry.edoc2.com:5000/edoc2v5/rabbitmq\_cluster:3.11.11-v1

3、用户指定img tag：docker push {用户指定的image tag}

# vim after\_edoc2\_rabbitmq\_cluster.yaml

修改 registry.edoc2.com:5000/edoc2v5/rabbitmq\_cluster:3.11.11-v1 为 {用户指定的image tag}

wq!保存退出

# vim edoc2\_rabbitmq\_cluster.yaml

修改 registry.edoc2.com:5000/edoc2v5/rabbitmq\_cluster:3.11.11-v1 为 {用户指定的image tag}

wq!保存退出

**二、通过手动脚本启动Rabbitmq集群操作：**

1、vim deploy\_rabbitmq.sh

#!/bin/bash

# 以下信息确认如果ssh免密不满足要求，则需要采用命令行纯手动命令行操作，目录创建纯手动部署模式，见此Rabbitmq-README.md文档三操作。

# 对应你部署环境namesapce命令（必填）

namespace=edoc2

# 部署三节点的k8snode节点主机名 k8s-node01 k8s-node02 k8s-node03 必须替换自己的节点主机名（必填）

WorkNode01=k8s-node01

WorkNode02=k8s-node02

WorkNode03=k8s-node03

# node节点ssh 端口

SSH\_PORT=22

# 默认写死 60G pv pvc空间大小

pv\_volume=60Gi

# 产研要求写死，此处提供mq数据配置挂载点 /home/data/rabbitmq 自动拼接 创建三节点目录 /home/data/rabbitmq01 /home/data/rabbitmq02 /home/data/rabbitmq03

pv\_path="/home/data/rabbitmq"

# 写死参数无需修改

pv\_yaml="rabbitmq-pv.yaml"

2、执行命令操作：

（1）、source deploy\_rabbitmq.sh start

（2）、source deploy\_rabbitmq.sh stop

（3）、source deploy\_rabbitmq.sh restart

**三、纯手动执行命令行操作：**

1、准备工作操作命令：

（1）、第一个k8s middleware宿主机节点操作执行命令：

mkdir -p /home/data/rabbitmq01

（2）、第二个k8s middleware宿主机节点操作执行命令：

mkdir -p /home/data/rabbitmq02

（3）、第三个k8s middleware宿主机节点操作执行命令：

mkdir -p /home/data/rabbitmq03

2、master01节点操作rabbitmq yaml编排文件：

 替换{需要修改的}

1. 、sed -i "s#namespace: edoc2#namespace {需要修改的}#g" \*.yaml && sed -i "s#namespace: edoc2#namespace {需要修改的}#g" \*.sh
2. 、sed -i "s#edoc2.svc#{新namespace}.svc#g" rabbitmq-configmap.yaml

（3）、vim rabbitmq-pv.yaml

# 修改如下信息：

 storage: 60Gi # 默认60Gi 修改存储pv大小三处

 path: /home/data/rabbitmq01 # 默认/home/data/，一定要修改只需修改 /home/data/根路径 rabbitmq01 rabbitmq02 rabbitmq03禁止修改

wq! 保存退出

（3）、vim edoc2\_rabbitmq\_cluster.yaml

# 修改如下信息：

# storage: 60Gi # 默认60Gi 修改存储pv大小一处（同rabbitmq-pv.yaml storage大小保持一致！）

wq! 保存退出

（4）、vim after\_edoc2\_rabbitmq\_cluster.yaml

# 修改如下信息：

# storage: 60Gi # 默认60Gi 修改存储pv大小一处（同rabbitmq-pv.yaml storage大小保持一致！）

wq! 保存退出

3、master01节点操作手动定义打节点标签：

（1）、执行如下操作：

# 定义node宿主机名字

WorkNode01={你环境的中间件第一节点主机名}

WorkNode02={你环境的中间件第二节点主机名}

WorkNode03={你环境的中间件第三节点主机名}

kubectl label nodes $WorkNode01 node.rabbitmq.labels=rabbitmq-node01

kubectl label nodes $WorkNode02 node.rabbitmq.labels=rabbitmq-node02

kubectl label nodes $WorkNode03 node.rabbitmq.labels=rabbitmq-node03

kubectl label nodes $WorkNode01 node.labels.nodelabels=Middleware

kubectl label nodes $WorkNode02 node.labels.nodelabels=Middleware

kubectl label nodes $WorkNode03 node.labels.nodelabels=Middleware

4、master01节点操作，启动rabbitmq集群操作命令：

（1）、创建StorageClass sc

kubectl apply -f rabbitmq-sc.yaml

（2）、创建 PV

kubectl apply -f rabbitmq-pv.yaml

（3）、创建rabbitmq configmap

kubectl apply -f rabbitmq-configmap.yaml

（4）、创建rabbitmq k8s上的rbac通讯权限

kubectl apply -f rabbitmq-rbac.yaml

（5）、第一次启动 .erlang.cookie 无数据执行如下命令（只有.erlang.cookie无此文件，第一次组集群使用）

kubectl apply -f edoc2\_rabbitmq\_cluster.yaml

（6）、等待5分钟后确认rabbitmq第一次启动成功命令：

kubectl get pod -n {你自己的namespace} # 默认namespace 为 edoc2

返回信息如下；

NAME READY STATUS RESTARTS AGE

rabbitmq-0 1/1 Running 0 1m

rabbitmq-1 1/1 Running 0 2m

rabbitmq-2 1/1 Running 0 3m

确认均为 Running切 READY都为 1/1

（7）、再次执行命令切换模式应对高可用测试宕机无序列启动rabbitmq组件集群：

kubectl apply -f after\_edoc2\_rabbitmq\_cluster.yaml

重要信息：

集群rabbtmq启动成功，集群运行正常后，均使用 kubectl apply -f after\_edoc2\_rabbitmq\_cluster.yaml启动

5、master01节点操作，停止rabbitmq集群操作命令：

（1）、停止rabbitmq集群服务

kubectl delete -f after\_edoc2\_rabbitmq\_cluster.yaml

sleep 30

等待彻底停止

（2）、删除pvc操作命令：

namespace={你自己的namespace命名}

kubectl delete pvc rabbitmq-storage-rabbitmq-0 -n $namespace

kubectl delete pvc rabbitmq-storage-rabbitmq-1 -n $namespace

kubectl delete pvc rabbitmq-storage-rabbitmq-2 -n $namespace

sleep 3

（3）、删除configmap操作命令：

kubectl delete -f rabbitmq-configmap.yaml

（4）、删除rbac操作命令：

kubectl delete -f rabbitmq-rbac.yaml

sleep 1

（5）、删除pv操作命令：

kubectl delete -f rabbitmq-pv.yaml

sleep 1

（6）、删除sc操作命令：

kubectl delete -f rabbitmq-sc.yaml

再次启动回到第4步操作！！！